目录

[**REVISION HISTORY** 1](#_Toc32673482)

[1 消息格式定义 2](#_Toc32673483)

[2 数据类型约定 2](#_Toc32673484)

[3 串口设置 3](#_Toc32673485)

[4 下位机文件结构 3](#_Toc32673486)

[5 通讯指令格式 5](#_Toc32673487)

[5.1 “0xFA” ACK格式说明 5](#_Toc32673488)

[5.2 “0x01”读/写仪器信息 5](#_Toc32673489)

[5.3 “0x02”密码配对 6](#_Toc32673490)

[5.4 “0x03”读取设备当前运行状态 6](#_Toc32673491)

[5.5 “0X04”读取当前实验名称 7](#_Toc32673492)

[5.6 “0x05”读取当前实验运行数据 7](#_Toc32673493)

[5.7 “0x06”读取故障 8](#_Toc32673494)

[5.8 “0x07” 启停实验 8](#_Toc32673495)

[5.9 “0x0A”上位机下发文件 9](#_Toc32673496)

[5.10 “0x0B”上位机读取文件 9](#_Toc32673497)

[5.11 “0X0C”固件升级 10](#_Toc32673498)

**REVISION HISTORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision** | **Change Order** | **Description** | **By** | **Date** |
| **0.1** | **Initial** | **Initial version** | **Jason Guo** | **2020/02/14** |
| **0.2** |  | **修改文件结构** | **Jason Guo** | **2020/04/29** |
| **0.3** |  | **添加08指令—校准** | **Jason Guo** | **2020/05/09** |
| **0.3** |  | **添加电机相关指令** | **Jason Guo** | **2020/05/12** |
| **0.4** |  | **细化文件读取指令** | **Jason Guo** | **2020/05/31** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 消息格式定义

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **消息头** | | **消息长度** | | **消息数据** | | **校验和** | | **结束符** |
| **HEAD** | | **LEN** | | **CMD** | **PARA** | **CHKSUM** | | **END** |
| **消息** | **7D** | **7D** | **L1** | **L2** | **C** | **Pi** | **C1** | **C2** | **0D** |
| **长度** | **2** Bytes | | **2** Bytes | | **1** Bytes | **n** Bytes | **2** Bytes | | **1** Byte |

* 消息中的数据都为十六进制形式；
* 消息中数据统一采用低字节在前高字节在后的形式；
* 消息长度LEN= DATA+校验+结束符，消息参数PARA中i为0~n，当n为零时表示此消息不含消息参数；
* 校验和Checksum=消息头 + 消息长度 + DATA的每个字节累加
* 消息回复超时时间100ms

## 数据类型约定

* 数据按类型低位在前发送；
* 长度(Length)两个字节，范围0x0000~0xFFFF。

Uint8 1字节，无符号整形数

Int8 1字节，有符号整形数

Uint16 2字节，无符号整形数

Int16 2字节，有符号整形数

Uint32 4字节，无符号整形数

Int32 4字节，有符号整形数

Float32 4字节

Char[] 字符串，长度不定，以‘\0’字符结尾

## 串口设置

**波特率：115200**

**数据位：8bit**

**校验位：无**

**停止位：1**

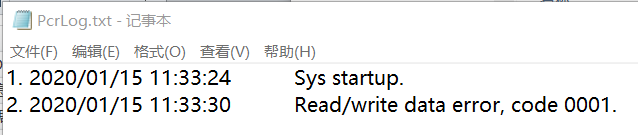
**流控：无**

## 下位机文件结构

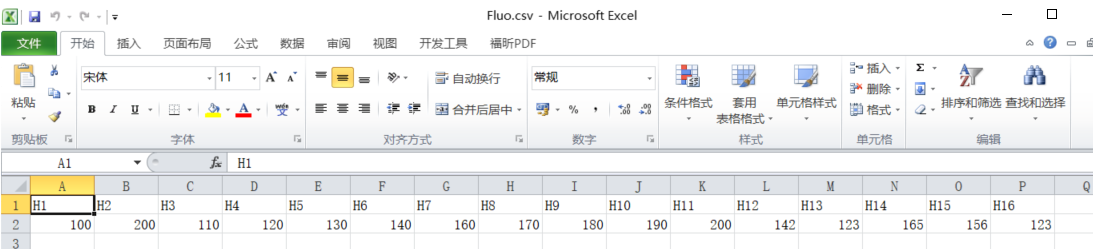
下位机数据全部存储于32MB的spiflash内，使用FATFS文件系统。所有文件可用过u盘拷贝，也可通过上位机读取。



1. 日志文件：PcrLog.txt位于根目录，大小限制在2MB内，超过会删除旧数据。格式示例如下：



1. Tmp文件夹：用于本地临时文件存储。
2. 实验名称文件夹：创建新的实验会创建对应的文件夹，文件夹名以用户输入的实验名命名。名称重复会覆盖数据；
3. 实验配置文件：config.ini位于根目录，保存本设备配置信息。
4. Json文件：位于实验名称文件夹内。保存本地实验配置数据。
5. 荧光数据文件：fluo.csv位于实验名称文件夹内。格式示例如下：



## 通讯指令格式

## “0XFA” ACK格式说明

ACK是指上位机与下位机指令执行结果确认的应答，格式如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **CMD** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

CMD—表示ACK的指令；

ACK—表示回复结果，有如下几种：

typedef enum {

ACK\_Fail = 0x00,//失败—表示动作执行失败

ACK\_OK = 0x01, //成功

ACK\_Error = 0x02, // 错误

ACK\_Busy = 0x03 // 忙 –表示下位机忙，等待100ms后重发

} AckType;

## “0x01”读/写仪器信息

写仪器信息时需要先发送0x02指令进行密码校验。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | | CHKSUM | | END |
| **7D** | **7D** | **L** | **01** | **Y1** | **Y2** | **Y3**  **Y3** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **01** | **Yn** | **X1** | **X2** | **0D** |

**参数说明：**

**Y1(Uint8)—表示执行读/写指令。01-读指令时，无Y3参数，参数Yn返回相应的信息内容；02-写指令时，参数Y2 Y3如下说明，Yn为ACK。**

**Y2(Uint8)—表示要执行的类型，见**type**。**

**Y3(Uint8[])—表示信息内容。**

|  |  |  |
| --- | --- | --- |
| ***Y2 (****type****)*** | ***Y3 (信息内容)*** | ***备注*** |
| *IDX\_PRODUCT\_TYPE = 0x01* | ”PCR16” | 产品型号 |
| *IDX\_PRODUCT\_SN = 0x02* | "202002141122" | 产品序列号 |
| *IDX\_MAINBOARD\_FWVer =0x03* | “V1.0.2R” | 主板固件版本号 |
| *IDX\_MAINBOARD\_PCBVer=0x04* | “V1.0.2” | 主板PCB版本号 |
| *IDX\_MAINBOARD\_PCBAVer= 0x05* | “V1.5” | 主板PCBA版本号 |
| *IDX\_MAINBOARD\_SN = 0x06* | “xxxxxxx” | 主板序列号 |

## “0x02”密码配对

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **02** | **Y**  **Y2**  **Y3** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **02** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

**Y(Uint8[])**—执行某些操作时需要密码匹配。密码；**PASS\_WORD[6] = {0x38,0x32,0x38,0x30,0x30,0x35};**

**ACK(Uint8 )—表示应答信息。**

## “0x03”读取设备当前运行状态

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **03** | **/**  **Y2**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **03** | **Y1** | **Y2** | **X1** | **X2** | **0D** |

参数说明：

**Y1(Uint8)**—运行状态。有如下几种：

typedef enum {

DevState\_Idle=0,//空闲

DevState\_Running=1,//实验中

DevState\_Pause=2,//实验暂停

DevState\_Standby=3,//休眠

DevState\_Error=4,//故障

DevState\_Debug=5,//调试中

}devstate;

**Y2(Uint8)**—设备子状态。与**Y1**对应关系如下：

仅当**Y1 = 1**“实验中”，**Y2**表示如下：

typedef enum {

DevSubState\_TempUp=0,//温度上升中

DevSubState\_TempDown=1,//温度下降中

DevSubState\_TempKeep=2,//温度保持中

DevSubState\_Fluo=3,//荧光采集中

DevSubState\_Unkown=0xFF,//未知

}devsubstate;

**Y1** 其它状态，**Y2**=0xFF无意义。

## “0X04”读取当前实验名称

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **04** | **/** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **04** | **Y1**  **Y2** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8[]**)—下位机当前实验名称。若无实验返回字符串“None”；

## “0x05”读取当前实验运行数据

上位机发送该指令时间间隔500ms

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **05** | **/** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **05** | **Yn**  **Y2** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—高4位表示当前实验阶段，总共分Stage0 - Stage15；

低4位表示当前实验步骤，总共分Step0 – Step15;

**Y2**(**Uint8**)—表示当前实验阶段处于第几个循环，数据范围1-255；

**Y3**(**Uint8**)—表示当前通道；

**Y4**(**Int16**)—热盖温度，精确到0.01℃；

**Y5**(**Int16**)—模块温度1，精确到0.01℃；

**Y6**(**Int16**)—模块温度2，精确到0.01℃；

**Y7**(**Uint16**)—当前步骤运行时间，单位秒；

**Y8**(**Uint8**)—实时荧光数据个数。**Y8** = 0表示无数据；当设备型号为PCR16，**Y8** = 16。设备型号为PCR48，**Y8** = 48，以此类推；

**Yn**(**Uint16**)—实时荧光数据，长度由**Y8**决定，**Y8** = 0，无此参数。

注：

1. 若当前无实验运行，全部返回0。
2. Y1或者Y2变化时，荧光数据Yn才会变化，否则一直保持。

## “0x06”读取故障

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **06** | **/** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **06** | **Y1** | **Y2** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—故障个数；

**Y2(Uint32)**—故障内容。1有效。详细如下：

|  |  |  |
| --- | --- | --- |
| Bit位 | 故障内容 | 错误代码 |
| 0 | 数据读写异常 | 0001 |
| 1 | 电机复位失败（无零点信号） | 0002 |

注：若当前无实验运行，全部返回0。

## “0x07” 启停实验

上位机设置下位机实验启停。设置之前需要下发实验参数，下发指令见0x0A。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **07** | **Y1** | **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **07** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8[]**)—实验名称。

**Y2**(**Uint8**)—要写入的实验状态。

enum {

SetDevState\_None=0,//无动作

SetDevState\_Running=1,//启动实验

SetDevState\_Pause=2,//暂停实验

SetDevState\_Stop=3,//停止实验

};

## “0x08” 执行校准

执行校准过程，设备当前运行状态为DevState\_Debug=5，结束后运行状态为DevState\_Idle=0。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **08** | **Y1**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **08** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—校准类型。

enum {

Cali\_HoleTempFluo=0,//空孔荧光

Cali\_HolePostion = 1,//孔位置校准

};

## “0x09” 读取校准结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **09** | **Y1**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **09** | **Y2**  **Y2** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—校准类型。

enum {

Cali\_HoleTempFluo=0,//空孔荧光

Cali\_HolePostion = 1,//孔位置校准

};

**Y2**(**Uint8[]**)—校准结果。根据校准类型Y1变化：

当Y1==0, Y2(**Uint32**)表示空孔荧光最小值(**Uint16**)和最大值(**Uint16**)；

当Y1==0, Y2(**Uint8[]**)表示孔位置(**Uint16**)

## “0x0A”上位机下发文件

上位机可下发文件到下位机，下位机会将文件保存在spiflash指定文件夹：/tmp

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **0A** | **T** | **Yn** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **0A** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

**T(Uint8)**—表示操作类型，包含类型如下：

typedef enum {

TYPE\_Query = 0x01, // 查询状态

TYPE\_Start = 0x02, // 传输启动

TYPE\_Transmit = 0x03, // 传输文件内容

TYPE\_End = 0x04 // 传输结束

} UpdateType;

参数 **ACK(Uint8)** —表示从机应答时的状态。参考5.1 ACK格式。

传输过程描述如下：

1. 发送操作类型 TYPE\_Start，表示启动发送文件。**Yn**(**Uint8[]**) 最大36字节:
2. **Y0**-**Y3** — 4字节表示整个文件的字节数；
3. 其它表示文件路径（最大32字节。该文件要保存在下位机的路径，若下位机不存该路径，会创建）；
4. 发送操作类型 TYPE\_Transmit，表示传输数据，**Yn**(**Uint8**[]) 表示数据，最大数据长度为 1KB；
5. 发送操作类型 TYPE\_End，表示发送完成，**Yn** (**Uint32**)表示 整个文件的校验和(所有字节的累加和)；
6. 发送操作类型 TYPE\_Query，用于查询更新情况，无**Yn**参数;

## “0x0B”上位机读取文件

上位机可以读取下位机flash中存储的文件。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **0B** | **T**  **T** | **Xn** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **0B** | **Yn**  **Y2** | **X1** | **X2** | **0D** |

参数说明：

**T(Uint8)**—表示操作类型，包含类型如下：

typedef enum {

TYPE\_Query = 0x01, // 查询状态

TYPE\_Start = 0x02, // 传输启动

TYPE\_Transmit = 0x03, // 传输文件内容

TYPE\_End = 0x04 // 传输结束

} UpdateType;

传输过程描述如下：

1. 发送操作类型TYPE\_Query，表示查询下位机状态，按照5.1 ACK格式回复。
2. 发送操作类型 TYPE\_Start，表示启动读取文件，**Xn**(**Uint8[]**) 表示读取文件路径（最大32字节字符串）；**Yn**(**Uint8[]**) 6字节，前4字节表示文件大小，后2字节表示分几个序列传输，每个序列最大传输1KB。当读取文件路径不存在时，**Yn**为0。
3. 发送操作类型 TYPE\_Transmit，表示请求传输数据，无**Xn**参数。**Yn**(**Uint8[]**)前两个字节表示序列号，其它表示数据内容。传输过程若下位机忙，将按照5.1 ACK格式回复ACK\_Busy。
4. 发送操作类型 TYPE\_End，表示接收完成，**Xn** (**Uint32**)表示整个文件的校验和(所有字节的累加和)，按照5.1 ACK格式回复**。**

## “0X0C”固件升级

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **0C** | **Y1** | **Y2**  **Y** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **0C** | **ACK** | **X1** | **X2** | **0D** |

说明：

**Y1(Uint8)**—表示升级对象，0x01—表示升级主板程序。

**Y2(Uint8)**—表示操作类型，包含类型查看 UpdateType ：

typedef enum {

TYPE\_JumpToIap = 0X05////jump to IAP

TYPE\_JumpToApp =0x06//jump to APP

} UpdateType;

Firmware 更新过程：

1. 发送操作类型TYPE\_JumpToIap，强制跳转至IAP模式。若从机当前处于app，反馈ACK\_OK并且跳转至IAP；若从机当前已经处于IAP，反馈ACK\_OK；
2. 传输过程见0x0A 下发文件指令；
3. 发送操作类型 TYPE\_JumpToApp，用于退出IAP，启动正常APP。

## “0x0D” LED灯控制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **0D** | **Y1** | **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **0D** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—LED灯类型，0--蓝色灯，1—绿色灯，2—所有灯；

**Y2**(**Uint8**)—LED灯开关控制，0—灭，1—点亮。

## “0x0E” 电机复位

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **0E** | **/**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **0E** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

无参数

## “0x0F” 移动电机到绝对位置

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **0F** | **Y1**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **0F** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint16**)—目标位置，精确到0.1mm

## “0x10” 查询电机状态

移动电机到绝对位置

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **10** | **/**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **10** | **Y1** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—电机状态反馈。

enum eMotorState {

MotorState\_Stop = 0, // Motor State:stop

MotorState\_Run = 1, // Motor State:run

}

## “0x11” 回读电机位置

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **11** | **/**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **11** | **Y1**  **ACK** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint16**)—电机当前位置，精确到0.1mm

## “0x12” 读取下位机存储信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **12** | **Y1**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **12** | **Y2** | **Y3** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—下位机存储类型。0x00-内存，0x01-flash

**Y2**(**Uint32**)—剩余空间。

**Y3**(**Uint32**)—总空间。

## “0x13” 温度设置

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | | CHKSUM | | END |
| **7D** | **7D** | **L** | **13** | **Y1** | **Y2** | **Y3** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | CHKSUM | | END |
| **7D** | **7D** | **L** | **FA** | **13** | **ACK** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—下位机模块类型。0x00—模块温度，0x01—热盖温度；

**Y2**(**Uint8**)—启动/停止温控。0x00—停止温控，0x01—启动温控；

**Y3**(**Uint16**)—温度值，精确到0.01度。

## “0x14” 读取温度值

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **14** | **Y1**  **Y2** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **14** | **Y2**  **Y3** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—下位机模块类型。0x00—模块温度，0x01—热盖温度。

**Y2**(**Uint16**)—温度值，精确到0.01度。

DEBUG指令

## “0xE0”设置PID参数

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | | | | CHKSUM | | END |
| **7D** | **7D** | **L** | **E1** | **Y1** | **Y2** | **Y3** | **Y4** | **X1** | **X2** | **0D** |

从机回复的数据格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | | LEN | MSG | PARA | CHKSUM | | END |
| **7D** | **7D** | **L** | **E1** | **ACK**  **Y3** | **X1** | **X2** | **0D** |

参数说明：

**Y1**(**Uint8**)—下位机模块类型。0x00—模块PID，0x01—热盖PID；

**Y2**(**float**)—PID参数中的P值；

**Y3**(**float**)—PID参数中的I值；

**Y4**(**float**)—PID参数中的D值；

**ACK**(**Uint8**)—表示应答信息。